Top-down Segmentation of Non-rigid Visual Objects using Derivative-based Search on Sparse Manifolds

Jacinto C. Nascimento
Instituto de Sistemas e Robótica
Instituto Superior Técnico
Lisboa, Portugal

Gustavo Carneiro
Australian Centre for Visual Technologies
The University of Adelaide
Adelaide, Australia

Abstract

The solution for the top-down segmentation of non-rigid visual objects using machine learning techniques is generally regarded as too complex to be solved in its full generality given the large dimensionality of the search space of the explicit representation of the segmentation contour. In order to reduce this complexity, the problem is usually divided into two stages: rigid detection and non-rigid segmentation. The rationale is based on the fact that the rigid detection can be run in a lower dimensionality space (i.e., less complex and faster) than the original contour space, and its result is then used to constrain the non-rigid segmentation. In this paper, we propose the use of sparse manifolds to reduce the dimensionality of the rigid detection search space of current state-of-the-art top-down segmentation methodologies. The main goals targeted by this smaller dimensionality search space are the decrease of the search running time complexity and the reduction of the training complexity of the rigid detector. These goals are attainable given that both the search and training complexities are function of the dimensionality of the rigid search space. We test our approach in the segmentation of the left ventricle from ultrasound images and lips from frontal face images. Compared to the performance of state-of-the-art non-rigid segmentation system, our experiments show that the use of sparse manifolds for the rigid detection leads to the two goals mentioned above.

1. Introduction

The top-down segmentation of non-rigid visual objects based on machine learning approaches [4, 9, 28, 29, 30, 31] has been traditionally addressed by dividing it into two procedures that are run in the following sequence: 1) rigid detection and 2) non-rigid segmentation. The main reason for the existence of a rigid detection step is the reduction of the training complexity and the search running time complexity. More specifically, assuming that the explicit representation of the segmentation contour consists of S 2-D points (usually, $S > 10$), the complexity of the exhaustive search would be $O(K^{2S})$, where K denotes the number of samples in each of the $2S$ dimensions. The introduction of the rigid detection procedure allows for a significant reduction of K during the non-rigid segmentation by constraining the search for the visual object borders within a small window around the output produced by the rigid detector. Usually, the rigid detection finds the center, scale and orientation of the visual object by searching for it in a parameter space of r dimensions, where $r << S$ (note that throughout the paper, r represents a variable that indicates the dimensionality of the rigid detection space). As a result, the rigid detection approach becomes the dominant procedure in terms of running time complexity, which is function of r.

Another issue addressed by the introduction of a rigid detection approach is the alleviation of the need of a large and complete training set. Nevertheless, the rigid detector still runs in an r-dimensional space, which means that the detector complexity is also function of r, and the training set available rarely provides enough information for a robust training. The usual solution to get around this issue consists of generating artificial positive and negative training samples by randomly perturbing the rigid parameters of the annotated data [25]. This random perturbation is commonly drawn from some simple probability density function (e.g., Gaussian distribution) learned from the annotations. The issue with this approach is that the actual training data distribution is unlikely to follow simple probability density functions, so this process is probably unnecessarily increasing the complexity of the learning process by adding artificial samples that may not be plausible in practice.

In this paper, we propose the use of sparse manifolds [18] with low intrinsic dimensionality for the rigid detection stage in non-rigid top-down visual segmentation methodologies [4, 5, 9, 30, 31]. The intrinsic low dimensionality of sparse manifolds decreases the search running time complexity of the current state-of-the-art rigid detection approaches aforementioned. Moreover, by restricting the positive and negative samples to lie in the learned low-dimensional sparse manifold, it is possible to reduce significantly the need for additional artificial positive and negative samples during the training process, and at the same

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.256

time guarantee that the additional samples are more likely to happen in practice. Consequently, this produces less complex and faster training processes. We test our approach in the challenging problems of left ventricle (LV) endocardial segmentation from ultrasound images and of lip segmentation from frontal face images. In the experiments, we produce competitive segmentation results with significant running time complexity reduction using a sparse manifold of two dimensions, which represents a substantial reduction from the original five dimensional rigid search space usually found in current approaches. Furthermore, we show that the training process is robust to the reduction of the number of additional artificial positive and negative samples, which indicates that smaller training sets can be used in our framework without affecting the segmentation accuracy. Also note that these smaller training sets result in less complex and faster training processes.

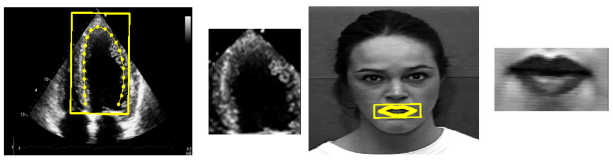
2. Related Work

The use of machine learning techniques to solve top-down non-rigid segmentation problems has been intensively investigated in the past few years. Particularly challenging problems in this domain are the segmentation of the left ventricle (LV) of the heart from ultrasound images [20] (see Figures 1(a) and 5), and the segmentation of the lip border from frontal face images [24] (see Figures 1(b) and 6). There have been several approaches to solve this problem using machine learning techniques, such as the boosting classifier for rigid detection followed by nearest neighbor search for the non-rigid segmentation [9, 30, 31], and the deep belief network approach for the rigid detection and the non-rigid segmentation [4, 3, 5]. However, none of these approaches attempt to reduce the dimensionality of the rigid search space using gradient-based search methods on manifolds.

Gradient-based search methods on manifolds have been recently investigated by Helmke et al. [11], who propose a new optimization approach for the essential matrix computation with the use of Gauss-Newton iterations on a manifold in order to reduce the computational effort. Hüper et al. [14] also elaborate a numerical optimization of a cost function defined on a manifold. In the same research line, Newton's method is applied along the geodesics and variants of projections are proposed where the optimization strategies take advantage of the manifold structure[1, 7, 23]. Our approach represents an application of such gradient-based search methods in the problem of top-down non-rigid segmentation with the specific goals of reducing the search running time and the training complexity.

3. Non-rigid Top-down Segmentation Problem Definition

Given an image containing the sought visual object, our goal is to produce a non-rigid segmentation using a matrix



(a) (b)

Figure 1. Application of the transformation $\mathbf{A_t}$ to the window enclosing the sought segmentation contour for the case of (a) left ventricle segmentation, and (b) lip segmentation. Both figures depict the explicit segmentation contour with the rectangular window (left panel) and zoomed in image of the visual information within the window (right panel). Note that the images on the right panels are the ones used by the rigid classifier $p(\mathbf{t}|I,\mathcal{D})$ in (2).

$\mathbf{S} \in \mathbb{R}^{2\times S}$ of S 2-D points, which is the explicit representation of the segmentation contour. Assume the availability of a training set, represented by $\mathcal{D} = \{(I,\mathbf{S})_j\}_{j=1}^{|\mathcal{D}|}$, containing training images $I_j : \Omega \rightarrow [0, 255]$ and the respective manual annotations $\mathbf{S}_j$, where Ω denotes the image lattice. The segmentation is achieved using the following function:

$$\mathbf{S}^* = \int_{\mathbf{S}} \mathbf{S} p(\mathbf{S}|I,\mathcal{D}) d\mathbf{S}. \quad (1)$$

The high dimensionality of $\mathbf{S}$ makes the computation of (1) intractable, and the usual solution to alleviate the problem is the introduction of an intermediate problem that can be solved in lower dimensionality, where the solution is used to constrain the optimization (1). This intermediate problem involves the use of a hidden variable $\mathbf{t} \in \mathbb{R}^r$, with $r << (2 \times S)$, as follows [4, 9, 30, 31]:

$$p(\mathbf{S}|I,\mathcal{D}) = \int_{\mathbf{t}} p(\mathbf{t}|I,\mathcal{D}) p(\mathbf{S}|\mathbf{t},I,\mathcal{D}) d\mathbf{t}. \quad (2)$$

In practice, the variable $\mathbf{t}$ is used to transform linearly the coordinates of a window that encloses the segmentation contour (see Fig. 1). This linear transform is obtained from the variable $\mathbf{t}$ as follows: $\mathbf{A_t} = f(\mathbf{t})$, where $\mathbf{A_t} \in \mathbb{R}^{3\times 3}$ [4, 9, 30, 31][1]. Then the term $p(\mathbf{t}|I,\mathcal{D})$ in (2) represents the rigid detection classifier that outputs the probability of having the sought visual object within the boundaries of the window transformed by $\mathbf{t}$. The term $p(\mathbf{S}|\mathbf{t},I,\mathcal{D})$ in (2) is the non-rigid segmentation classifier denoted by the probability of finding the contour $\mathbf{S}$ in image I given the value of $\mathbf{t}$. That is, $\mathbf{t}$ constrains the search space of $\mathbf{S}$ to be within the image window defined by $\mathbf{t}$.

Assuming that the original rigid search space represented by the variable $\mathbf{t}$ has dimension $r = R$, the main objective of this paper is the introduction of a new space for $\mathbf{t}$ with dimension $r = M < R$, based on a sparse manifold.

[1] For example, suppose $\mathbf{t} = [x, y, \vartheta, \nu_x, \nu_y]$ denotes a transformation comprising a translation x and y, rotation ϑ, and non-uniform scaling ν_x and ν_y; then $f(\mathbf{t}) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \nu_x & 0 & 0 \\ 0 & \nu_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

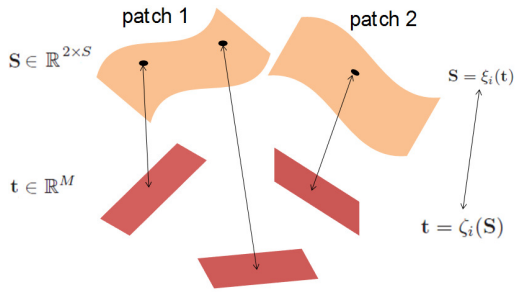


Figure 2. Partition of the manifold into patches (top) and the corresponding tangent hyperplanes (bottom). The arrows illustrate the mappings back and forth between the patches and the hyperplanes. The black dots are the annotations from which a low dimensional representation is built.

4. Sparse Manifolds

The intuition of using manifolds for reducing the dimensionality of $\mathbf{t}$ is centered on the idea that the training annotations $\mathbf{S}_j$ can be confidently represented in a manifold $\mathcal{M}$ by the respective lower dimensional variable $\mathbf{t}_j$. For learning such manifold, we follow the *Gaussian Processes Multiple Dynamical Models* (GP-MLM) [18] implementation. GP-MLM is a local method that finds multiple representations of the manifold, valid in different regions. The advantages of GP-MLM compared to other approaches [26, 27] are: 1) the data partitioning into several local models allows us to make less restrictive assumptions about the topology of the manifold; 2) the coordinates of each local model are found by simply projecting the data onto previously estimated tangent subspaces, instead of performing costly iterative optimizations (e.g., GP-LVM [19]); and 3) it allows for an elegant way to perform model selection approaches, which increases the efficiency of GP-MLM.

The manifold learning strategy consists of the following input/output (see Fig. 2 for an illustration):

- **Input**: training samples $\mathbf{S}_j$, $j = 1, ..., |\mathcal{D}|$,
- **Output**: (i) intrinsic manifold dimension $M < R << S$; (ii) partition the manifold $\mathcal{M}$ into overlapping patches $\mathcal{P}_i \subset \mathcal{M}$, $i \in \{1, ..., |\mathcal{P}|\}$; (iii) tangent hyperplanes $T_{\mathcal{P}_i}$ for each patch; and (iv) forward and backward mappings between patches and tangent hyperplanes, *i.e.* the *charts* $\mathbf{t} = \zeta_i(\mathbf{S})$ and *parameterizations* $\mathbf{S} = \xi_i(\mathbf{t}) + \boldsymbol{\omega}$, in which $\boldsymbol{\omega}$ represents a zero mean Gaussian observation noise.

According to GP-MLM, each patch $\mathcal{P}_i$ is represented by $|\mathcal{P}_i|$ samples drawn from the training set $\mathcal{D}$. These samples are then used to form the matrix $\mathbf{T}_i = [\mathbf{t}_{i,1}, ..., \mathbf{t}_{i,|\mathcal{P}_i|}] \in \mathbb{R}^{M\times|\mathcal{P}_i|}$, where the $|\mathcal{P}_i|$ points belonging to patch $\mathcal{P}_i$ are known as the *patch member* points, and in general $|\mathcal{P}_i| \neq |\mathcal{P}_j|$ for $i \neq j$.

One of the innovations of this paper is the execution of the function in (2) directly on the manifold $\mathcal{M}$. This is accomplished by performing the optimization process in each of the low dimensional patches $\mathcal{P}_i$ with initial guesses taken from the patch member points $\mathbf{t}_{i,j} = \zeta_i(\mathbf{S}_{i,j})$, for $j = 1, ..., |\mathcal{P}_i|$. The main issue with this approach is that GP-MLM may provide a relatively large number of patch members, especially if the sought object presents significant variations in terms of location, scale and rotations in the training image sequence. Consequently, this large number of patch members results in a large number of initial guesses, which decreases the efficiency of the search process. In order to reduce the number of initial guesses, we propose a patch member selection procedure, where the goal is to select a subset of the columns of $\mathbf{T}_i = [\mathbf{t}_{i,1}, ..., \mathbf{t}_{i,|\mathcal{P}_i|}]$ that preserves enough information about the chart ζ_i. This selection will provide automatically a set of *landmarks*. To solve this problem we start by converting the non-linear regression problem $\mathbf{S} = \xi_i(\mathbf{t}) + \boldsymbol{\omega}$ into a linear regression by off-loading the non-linearity onto a kernel [21]. The linear conversion takes the following matrix form

$$\boldsymbol{\Theta}^\top = \mathbf{K}\mathbf{B} + \mathbf{W}, \tag{3}$$

where $\boldsymbol{\Theta} = [(\pi_1\mathbf{S}_{i,1}, \pi_2\mathbf{S}_{i,1})^\top, ..., (\pi_1\mathbf{S}_{i,|\mathcal{P}_i|}, \pi_2\mathbf{S}_{i,|\mathcal{P}_i|})^\top] \in \mathbb{R}^{2S\times|\mathcal{P}_i|}$ (with $\pi_1 = [1, 0]$ and $\pi_2 = [0, 1]$), $\mathbf{K}$ is a $|\mathcal{P}_i| \times |\mathcal{P}_i|$ symmetric positive semi-definite matrix with elements $k_{ij} = \exp(-\|\mathbf{t}_i - \mathbf{t}_j\|^2/2\sigma_K)$, $\mathbf{W} \in \mathbb{R}^{|\mathcal{P}_i|\times 2S}$ with each row being a realization of $\boldsymbol{\omega}$, and $\mathbf{B} \in \mathbb{R}^{|\mathcal{P}_i|\times 2S}$. This is however, difficult to solve since it leads to a *multiple measurement vectors* (MMV) problem [6], where it is difficult to impose sparsity in all columns simultaneously. To tackle this issue we re-formulate (3) as a single measurement vector problem (SMV) [22] leading to

$$\boldsymbol{\theta} = \mathbf{K}\boldsymbol{\beta} + \eta, \tag{4}$$

where $\boldsymbol{\theta}, \boldsymbol{\beta}, \eta \in \mathbb{R}^{|\mathcal{P}_i|}$ are column vectors, from which it is possible to use model selection based procedures. In this work, the j^{th} element θ_j of the vector $\boldsymbol{\theta}$, is the maximum principal angle between the tangent bundles $T_{\mathbf{S}_{i,j}}$ and $T_{\mathcal{P}_i}$, where $T_{\mathbf{S}_{i,j}}$ is the tangent subspace of $\mathbf{S}_{i,j}$ and $T_{\mathcal{P}_i}$ is the tangent subspace of $\mathcal{P}_i$ found by agglomerative soft clustering [18]. In effect, notice that this reformulation allows the representation of each annotation $\mathbf{S}_{i,j}$ by the element θ_j of the vector $\boldsymbol{\theta}$.

The model selection (to find the set of landmarks) is then achieved by finding a sparse solution to (4). The idea is to estimate $\boldsymbol{\beta}$ with the minimization of the following cost function:

$$\left\|\boldsymbol{\theta} - \mathbf{K}\boldsymbol{\beta}\right\| + \alpha\left\|\boldsymbol{\beta}\right\|_q, \tag{5}$$

where $\|\boldsymbol{\beta}\|_q$ is the ℓ_q norm of $\boldsymbol{\beta}$, and α controls the regularization term. To minimize (5), we follow the *least angle regression* (LARS) procedure [8], where the risk is a function of the number L_i of non-zero values in $\boldsymbol{\beta}$, forming the vector $\boldsymbol{\beta}_{L_i}$ (note that L_i denotes the number of landmarks

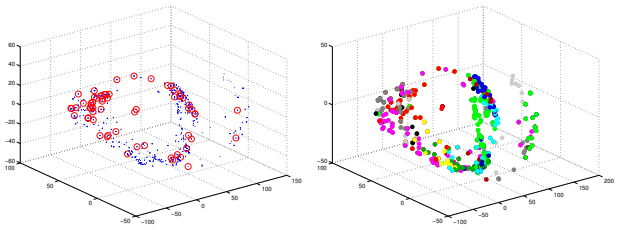


Figure 3. Manifold learning algorithm for the LV segmentation problem. The graph on the left shows the annotation points in blue and landmarks in red. From our experiments, a total of 1158 patch member points (blue dots) and 63 landmark points (circle red) are estimated. On the right graph, each annotation point is colored according to which patch it belongs, where from the experiments, 14 patches are estimated.

per patch $\mathcal{P}_i$). This risk minimization problem forms the set of landmarks for the i^{th} patch $\widetilde{\mathbf{T}}_i = [\mathbf{t}_1, ..., \mathbf{t}_{L_i}] \in \mathbb{R}^{M \times L_i}$ with $L_i << |\mathcal{P}_i|$, where the columns of $\widetilde{\mathbf{T}}_i \subseteq \mathbf{T}_i$ are the landmarks, which have the same indexes as the non-zero elements of $\boldsymbol{\beta}_{L_i}$. These landmarks will be the points used for initial guesses in the segmentation procedure, and in general we have $L_i \neq L_j$ for $i \neq j$.

Fig. 3 illustrates the result of our manifold learning algorithm on the LV segmentation problem (see Section 8), where each patch $\mathcal{P}_i$ contains a set of the patch-member and landmark points. On the left, the blue dots are the annotations after PCA reduction (the first three components are shown), and the red circles indicate the landmarks. On the right we can see the patches, where each color represents a different patch.

5. Training and Inference on the Sparse Manifold

The rigid detection classifier in (2) is modeled by the parameter vector γ_{MAP} (learned with a maximum a posteriori learning algorithm), which means that $p(\mathbf{t}|I, \mathcal{D})$ is represented by $p(\mathbf{t}|I, \gamma_{\text{MAP}})$. The estimation of γ_{MAP} is based on a set of training samples taken from the set of *patch member* points $\mathbf{t}_{i,j} = \zeta_i(\mathbf{S}_{i,j})$ (for $j \in \{1, ..., |\mathcal{P}_i|\}$) of each learned patch $\mathcal{P}_i$ (for $i \in \{1, ..., |\mathcal{P}|\}$) estimated by the manifold learning algorithm. Specifically, the generation of positive and negative samples involves the following steps: 1) estimate the contour in the original image space from the landmark, $\hat{\mathbf{S}}_{i,j} = \zeta_i^{-1}(\mathbf{t}_{i,j})$; 2) find the transformation matrix $\mathbf{A}_{\mathbf{t}_{i,j}}$ of the image window enclosing the segmentation contour $\hat{\mathbf{S}}_{i,j}$ produced in step (1).

For training the classifier, the sets of positives and negatives are formed by sampling a distribution of the patch members $\mathbf{t}_{i,j}$. The distribution in patch $\mathcal{P}_i$ is defined by

$$\text{Dist}(\mathcal{P}_i) = U(range(\mathbf{T}_i)), \tag{6}$$

where $U(range(\mathbf{T}_i))$ is the uniform distribution such that $range(\mathbf{T}_i) = [\max_{row}(\mathbf{T}_i) - \min_{row}(\mathbf{T}_i)] \in \mathbb{R}^M$ with $\mathbf{T}_i$ defined in Sec. 4, denoting a matrix with the patch members $\mathbf{t}_{i,j} \in \mathcal{P}_i$ in its columns and the functions $\max_{row}(\mathbf{T}_i) \in \mathbb{R}^M$ and $\min_{row}(\mathbf{T}_i) \in \mathbb{R}^M$ representing, respectively, the maximum and minimum row elements of the matrix $\mathbf{T}_i$. The positive and negative sets are respectively generated as follows:

$$\begin{aligned} \mathcal{T}_+(i,j) &= \{\mathbf{t} | \mathbf{t} \sim \text{Dist}(\mathcal{P}_i), d(\mathbf{t}, \mathbf{t}_{i,j}) \prec \mathbf{m}_i\} \\ \mathcal{T}_-(i) &= \{\mathbf{t} | \mathbf{t} \sim \text{Dist}(\mathcal{P}_i), d(\mathbf{t}, \mathbf{t}_{i,j}) \succ 2 \times \mathbf{m}_i\,, \\ &\qquad \forall j \in \{1, ..., |\mathcal{P}_i|\} \end{aligned} \tag{7}$$

where

$$\mathbf{m}_i = range(\mathbf{T}_i) \times \kappa \tag{8}$$

represents the margin between positive and negative cases with $\kappa \in (0, 1)$ defined as a constant, the $\prec$ and $\succ$ are the element wise "less than" or "greater than" operators between two vectors, and $d(\mathbf{t}, \mathbf{t}_{i,j}) = |\mathbf{t} - \mathbf{t}_{i,j}| \in \mathbb{R}^M$ is the dissimilarity function in (7), where $|.|$ returns the absolute value of the vector $\mathbf{t} - \mathbf{t}_{i,j}$. Note that the randomly generated parameter $\mathbf{t}$ in (7) is projected to the patch $\mathcal{P}_i$ in order to guarantee that it belongs to the manifold. Basically, (7) generates positive samples that are relatively close to patch member points and negative samples that are sufficiently far to all patch members.

Finally, the discriminative learning of the rigid classifier is achieved with the maximization of the following objective function [12]:

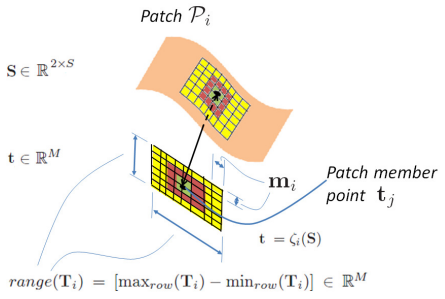
$$\begin{aligned} \gamma_{\text{MAP}} = \arg\max_{\gamma} \prod_{i=1}^{|\mathcal{P}|} \prod_{j=1}^{|\mathcal{P}_i|} \left[\prod_{\mathbf{t} \in \mathcal{T}_+(i,j)} p(\mathbf{t}|I, \gamma) \right] \\ \times \left[\prod_{\mathbf{t} \in \mathcal{T}_-(i)} (1 - p(\mathbf{t}|I, \gamma)) \right]. \end{aligned} \tag{9}$$

Fig.4(a) displays the training process explained in this section, where the positive samples are extracted from the green region, and the negative samples are drawn from the yellow region. The parameters λ of the non-rigid classifier in (2) are learned similarly with the following optimization:

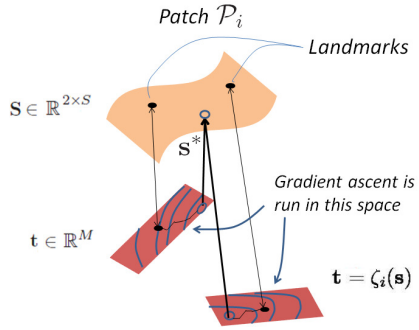
$$\lambda_{\text{MAP}} = \arg\max_{\lambda} \prod_{i=1}^{|\mathcal{P}|} \prod_{j=1}^{|\mathcal{P}_i|} p(\mathbf{S}_{i,j}|\mathbf{t}_{i,j}, I, \lambda), \tag{10}$$

where $p(\mathbf{S}|\mathbf{t}, I, \mathcal{D})$ in (2) is represented by $p(\mathbf{S}|\mathbf{t}, I, \lambda_{\text{MAP}})$.

The inference procedure to generate the segmentation contour takes a test image represented by I, and uses the sparse manifold to produce the contour $\mathbf{S}^* \in \mathbb{R}^{2 \times S}$ in (1). It is important to mention that this inference procedure uses each *landmark* $\mathbf{t}_{i,j}$ (for $j \in \{1, ..., L_i\}$) from each learned patch $\mathcal{P}_i$ as initial guesses for a gradient ascent (GA) procedure [2] on the output of the classifier $p(\mathbf{t}|I, \gamma_{\text{MAP}})$ over the search parameter space on the manifold $\mathcal{M}$. Given



(a) Training



(b) Inference

Figure 4. Training (top) and inference (bottom) procedures (please see text for details).

that the initial guesses of the GA procedure come from the landmarks, we have $\mathbf{t}_{i,j}^{(0)} = \mathbf{t}_{i,j}$ (the superscript (n) for $n \in \{0..N\}$ represents the GA iteration index, with N denoting the maximum number of GA steps), and after N GA iterations, the final value for the search parameter is $\mathbf{t}_{i,j}^{(N)}$. Assuming that $p(\mathbf{t}) = p(\mathbf{t}|I, \gamma_{MAP})$, the GA algorithm uses the Jacobian $\nabla p(\mathbf{t}) = \left[\begin{array}{ccc} \frac{\partial p(\mathbf{t})}{\partial \mathbf{t}(1)} & \ldots & \frac{\partial p(\mathbf{t})}{\partial \mathbf{t}(M)} \end{array} \right]^\top$, which is computed numerically using central difference, with step size $\mathbf{m}_i$ (8), as follows:

$$\frac{\partial p(\mathbf{t})}{\partial \mathbf{t}(1)} = \frac{p(\mathbf{t} + [\mathbf{m}_i(1)/2, ..., 0]^\top) - p(\mathbf{t} - [\mathbf{m}_i(1)/2, ..., 0]^\top)}{\mathbf{m}_i(1)} \tag{11}$$

where the parameter for $\mathbf{t}(.)$ indicates the dimensionality index (i.e., $\mathbf{t}(1)$ denotes the first dimension of $\mathbf{t}$), and similarly for $\mathbf{m}_i(.)$. In (11), the parameter $\mathbf{t} \pm [\mathbf{m}_i(1)/2, ..., 0]^\top$ is projected to the patch $\mathcal{P}_i$ (*i.e.*, $\mathbf{S}_{i,j} = \xi_i(\mathbf{t}_{i,j})$) in order to guarantee that it belongs to the manifold $\mathcal{M}$. Once the GA process is over and the parameter $\mathbf{t}_{i,j}^{(N)}$ is reached for each landmark $\mathbf{t}_{i,j}$ of each patch $\mathcal{P}_i$

, and the contour $\mathbf{S}^*$ is estimated with a Monte-Carlo approximation of (1) as follows:

$$\mathbf{S}^* = \frac{1}{Z} \sum_{i=1}^{|\mathcal{P}|} \sum_{j=1}^{L_i} \mathbf{S} p(\mathbf{t}_{i,j}^{(N)}|I, \gamma_{\text{MAP}}) \times p(\mathbf{S}|\mathbf{t}_{i,j}^{(N)}, I, \lambda_{\text{MAP}}), \tag{12}$$

where Z is a normalization factor. Figure 4(b) shows the setting of the segmentation procedure, with the level sets representing the results of the rigid classifier $p(\mathbf{t}_{i,j}|I, \gamma_{\text{MAP}})$. Notice that the rigid search procedure is performed only in the low dimensional space of $\mathbf{t}$.

6. Search Complexity Reduction

The bottleneck of current top-down non-rigid segmentation methods based on machine learning techniques lies in the number of executions of the rigid classifier $p(\mathbf{t}|I, \gamma_{\text{MAP}})$ that runs in the intermediate space represented by the variable $\mathbf{t} \in \mathbb{R}^r$. For the complexity analysis below, assume that $K = \mathcal{O}(10^3)$ denotes the number of samples used in each dimension of this intermediate space. An exhaustive search in this r-dimensional space represents a running time complexity of $O(K^r)$, which is in general intractable for relatively small values of $r = R$ (note that $R \in \{4, 5\}$ in state-of-the-art approaches). The reduction of this running time complexity has been studied by Lampert et al. [15], who proposed a branch-and-bound approach that can find a global optimum in this rigid search space in $O(K^{r/2})$. Zheng et al. [30] proposed the marginal space learning that finds local optima using a coarse-to-fine approach, where the search space is recursively broken into spaces of increasing dimensionality (i.e., the search begins with one of the r dimensions, whose result is used to constrain the search in the space of two dimensions, until arriving at the space of r dimensions). Carneiro et al. [4] also proposed a local optima approach based on a coarse-to-fine derivative-based search that uses a gradient ascent approach in the space of r dimensions. In general, these last two methods provide a search complexity of $O(K + T \times K_{fine} \times r)$, where T is the number of scales (for the methods above, $T = 3$), and $K_{fine} << K$ (commonly, $K_{fine} = \mathcal{O}(10^1)$). Notice that our proposal has the potential to increase the efficiency of all approaches above because we reduce r from R to M. Moreover, with the use of L_i landmarks per patch $\mathcal{P}_i$, we avoid the expensive initial search of K points in the coarsest scale. Taking all this together, we have a final complexity of $O((\sum_i L_i) \times T \times r)$. Typically, we have $\sum_i L_i = O(10^1)$, so our approach leads to a complexity of $O(10 \times 3 \times r)$, which compares favorably to $O(10^3 + 3 \times 10 \times r)$ [4, 30] and $O((10^3)^{r/2})$ [15].

7. Databases and Implementation Details

Two databases are used in the experiments. For the LV segmentation problem, we use the database proposed by Nascimento et al. [17]. In this database, the training set contains 400 ultrasound images of the left ventricle of the heart (using the apical two and four-chamber views), which have been taken from 12 sequences (12 subjects with no overlap), where each sequence contains an average of 34 annotated frames. The test set contains two sequences, where each sequence has 40 annotated images (two subjects with no overlap). Note that all images in the training and test sets

have been annotated by a cardiologist. For the lip segmentation problem, we use the CohnKanade (CK+) database [16] of emotion sequences, where manual annotation is available. Among several emotions, we select the "surprise" sequences since they exhibit challenging lip boundary deformations. The training set has four sequences (four subjects with no overlap), consisting of 103 annotated training images, and the test set contains 10 sequences (ten subjects with no overlap), comprising 171 images. For both databases, the intersection between training and test sets is empty and the test set is used exclusively for testing. Finally, the dimensionality of the explicit representation for the LV contour is $S = 21$, and for the lip segmentation contour is $S = 40$.

The demonstration of the proposed derivative-based search on sparse manifolds is conducted using an extension of the method proposed by Carneiro et al. [4], where the coarse-to-fine rigid detector $p(\mathbf{t}|I, \gamma_{\text{MAP}})$ and non-rigid classifier $p(\mathbf{S}|\mathbf{t}, I, \lambda_{\text{MAP}})$ are based on deep belief networks (DBN) [12]. The extension consists of training and running the rigid classifier in the space defined by the sparse manifold described in Sec. 4. The parameter κ in (8), defining the training grid for sampling positive and negative examples, is set to $1/200$, similarly to [4]. For the LV segmentation problem, the manifold learning algorithm produces: (i) $|\mathcal{P}| = 14$ patches, with a total of 1158 patch member points and 63 landmark points, and (ii) $M = 2$ for the dimensionality of the rigid search space (i.e., this represents the intrinsic dimensionality of the manifold) . For the lip segmentation, the manifold learning produces: (i) $|\mathcal{P}| = 1$ patch with 120 patch member points and 3 landmark points, and (ii) $M = 2$ for the dimensionality of the rigid search space. It is worth mentioning that the original dimensionality of the rigid search space for current approaches is $R = 5$ (representing two translation, one rotation and two scale parameters) [4, 9, 31]. Finally, in order to estimate the robustness of our approach to small training sets we conduct an experiment on the LV database, where we vary the size of the set of positive samples as follows $|\mathcal{T}_+(i,j)| \in \{1, 5, 10\}$, and the size of negative samples as $|\mathcal{T}_-(i)| \in \{10, 50, 100\} \times |\mathcal{P}_i|$ (we added more additional negative samples due to the larger area occupied by the negative region). Our goal with this experiment is to study how the accuracy of the methodology varies with the size of the database. Note that for both databases, the training of the original algorithm in [4] used $|\mathcal{T}_+| = 10$, and $|\mathcal{T}_-| = 100$ per each image in the training set (*i.e.*, 10 additional positive samples and 100 negative samples per training image), which corresponds to the largest size of $|\mathcal{T}_+(i,j)|$ and $|\mathcal{T}_-(i)|$, defined in (7).

8. Experimental Setup and Results

In this section we present segmentation results of the proposed approach in the problems of LV and lip segmentation. The performance is evaluated in terms of contour accuracy (using several metrics commonly adopted in the literature), running time spent to perform the object segmentation, and the performance of the classifier as a function of $|\mathcal{T}_+(i,j)|$ and $|\mathcal{T}_-(i)|$, as described in Sec. 7.

The performance of the segmentation is assessed using the following error measures proposed in the literature for assessing the accuracy of segmentation approaches: (i) *Hausdorff* (HDF) [13], (ii) *mean absolute distance* (MAD) [31], (iii) *Jaccard* (JCD) [10], and (iv) *average* (AV) [17]. The performance of our approach is assessed by a quantitative comparison over the test sets with the following state-of-the-art methods (based on machine learning techniques) proposed in the literature for the LV segmentation problem: COM [9, 31], CAR1 [4], CAR2 [5]. For the lip segmentation, we compare the performance of our approach only with CAR1 [4] because that was the only one available for comparison in this problem. For both segmentation problems, we also compare the running times between our approach and CAR1 [4] (recall that our approach is an extension of CAR1).

Table 1 shows the quantitative comparison for the two test sequences of the LV segmentation problem, where the best values in each row are highlighted. Moreover, Table 1 shows how the performance of our approach is affected with the varying sizes of $\mathcal{T}_+(i,j)$ and $\mathcal{T}_-(i)$. For illustration purposes, Fig. 5 shows the best and worst segmentation results (using the segmentation method described in this paper) in the two test sequences with the corresponding Jaccard error and mean running time (in seconds) per sequence. Finally, we compare the running time figures of our approach with CAR1 [4]. The mean running time of our approach for both sequences is 2.37 seconds, while for CAR1 is 7.4 seconds. It is important to mention that these running time figures were obtained with unoptimized Matlab implementations.

For the lip segmentation, Tab. 2 shows a comparison between our approach and CAR1 [4], where the best values in each cell are highlighted. In these experiments, the training sets are fixed at $|\mathcal{T}_+(i,j)| = 10$ and $|\mathcal{T}_-(i)| = 100$. Fig. 6 shows a subset of the lip segmentation results produced by our approach, where each image corresponds to a different test sequence and the segmentation results (in red color) are superimposed with the ground truth (in green color). At the bottom of each image, it is displayed the error measure between the estimated and manual contours using *Jaccard* (JCD) [10] metric, and the running time spent in seconds (in parenthesis). The mean running time of our approach for all lip sequences is 2.62 seconds, while for the method in [4] is 11.8 seconds. Similarly to the LV segmentation problem, these running time figures were obtained using unoptimized Matlab implementations.

9. Discussion and Conclusions

The main conclusion that can be drawn from the experiments above is that the derivative-based search on sparse manifolds enables a significant improvement in terms of ef-

Table 1. Quantitative comparison in the test sequences of the LV segmentation problem. Each cell shows the mean and standard deviation of the respective error measure.

Test set 1						
measures	COM [9, 31]	CAR1 [4]	CAR2 [5]	$\lvert\mathcal{T}_+(i,j)\rvert = 1$ $\lvert\mathcal{T}_-(i)\rvert = 10$	$\lvert\mathcal{T}_+(i,j)\rvert = 5$ $\lvert\mathcal{T}_-(i)\rvert = 50$	$\lvert\mathcal{T}_+(i,j)\rvert = 10$ $\lvert\mathcal{T}_-(i)\rvert = 100$
HDF	20.48(1.19)	**19.18(2.28)**	20.60(2.60)	19.39(1.54)	20.15(1.83)	19.94(1.60)
MAD	11.40(3.19)	9.92(3.33)	**9.43(2.09)**	9.82(2.71)	10.39(2.99)	11.05(3.34)
JCD	0.21(0.04)	**0.17(0.05)**	0.18(0.06)	0.19(0.03)	0.21(0.03)	0.23(0.06)
AV	3.89(0.56)	**3.33(0.86)**	3.28(0.84)	3.88(0.70)	4.38(0.84)	4.85(1.46)
Test set 2						
HDF	**17.21(1.37)**	19.43(1.48)	19.88(1.88)	19.25(1.72)	19.39(2.47)	18.90(2.23)
MAD	18.16(6.08)	15.71(5.66)	17.74(5.50)	16.63(6.31)	16.17(6.72)	**15.53(6.17)**
JCD	0.19(0.03)	**0.16(0.04)**	0.17(0.02)	0.19(0.03)	0.20(0.04)	0.20(0.03)
AV	3.37(0.6)	**2.93(0.55)**	3.08(0.58)	3.76(0.66)	4.00(0.78)	3.82(0.60)

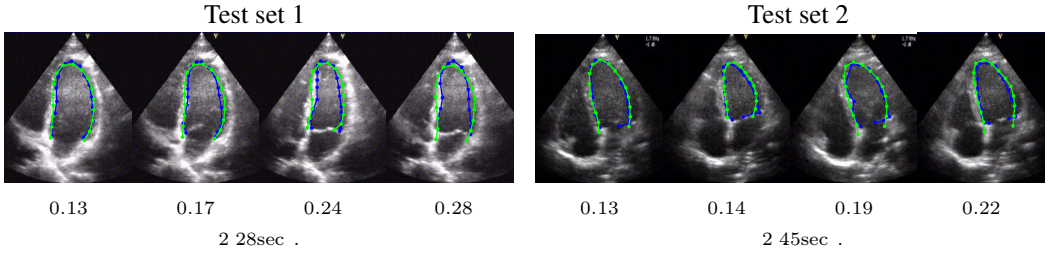


Figure 5. LV segmentation results on the test set. Left panel: best (two left snapshots) and worst (two right snapshots) segmentation results in the test sequence 1 with the correspondent Jaccard index and mean running time for each sequence. Right panel: the same for the test sequence 2. The ground-truth is in blue, and the segmentation estimated by our approach is in green.

ficiency without a negative impact in terms of non-rigid segmentation accuracy. Specifically, the proposed approach is around 3 to 4 times faster than the method CAR1 [4]. It is conceivable that similar efficiency improvements can be achieved in other similarly designed non-rigid segmentation approaches [9, 30, 31]. Also, we can see that for the problem of lip segmentation the use of sparse manifolds actually improves the accuracy. We believe that this improvement can be explained by the reduction of the noise that is generally achieved in spaces of smaller dimensionality. Finally, Tab. 1 also shows that our approach does not present statistically significantly different segmentation results with respect to training sets of varying sizes, which means that we can have fewer additional training samples and still obtain competitive results. The main advantage of this last observation is the fact that the training can be a lot faster with the use of fewer training samples.

The main open issues in this work are with respect to the model selection problem in the manifold learning algorithm, and the need of the intermediate space for the top-down segmentation methodology. Specifically, finding an optimal number of patches in the manifold represents a model selection problem, which we intend to investigate with the goal of selecting a parsimonious model that is capable of describing the observed data and generalizing to unobserved data. Finally, the need of computing the rigid transform $\mathbf{A}$ after finding $\mathbf{S}$ from the variable $\mathbf{t}$ can be questioned. We plan to extend this work by avoiding the computation of $\mathbf{A}$ altogether and use a non-rigid deformation to produce the input for the classifier directly.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemmanian geometry of Grassman manifolds with a view on algorithmic computation. In *Acta Applicandae Mathematicae*, volume 80, pages 199–220, 2004. 2

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. 4

[3] G. Carneiro and J. Nascimento. The fusion of deep learning architectures and particle filtering applied to lip tracking. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, pages 2065–2068. IEEE Computer Society, 2010. 2

[4] G. Carneiro and J. C. Nascimento. Multiple dynamic models for tracking the left ventricle of the heart from ultrasound data using particle filters and deep learning architectures. In *CVPR*, pages 2815–2822, 2010. 1, 2, 5, 6, 7, 8

[5] G. Carneiro, J. C. Nascimento, and A. Freitas. Robust left ventricle segmentation from ultrasound data using deep neural networks and efficient search methods. In *IEEE Int. Symp. on Biomedical Imaging, from nano to macro (ISBI)*, pages 1085–1088, 2010. 1, 2, 6, 7

[6] J. Chen and X. Huo. Sparse representation for multiple measurements vectors (MMV) in an over-complete dictionary. In *ICASSP*, 2005. 3

[7] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality consraints. In *SIAM J. Matrix Anal. Appl.*, volume 20, pages 303–353, 1998. 2

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004. 3

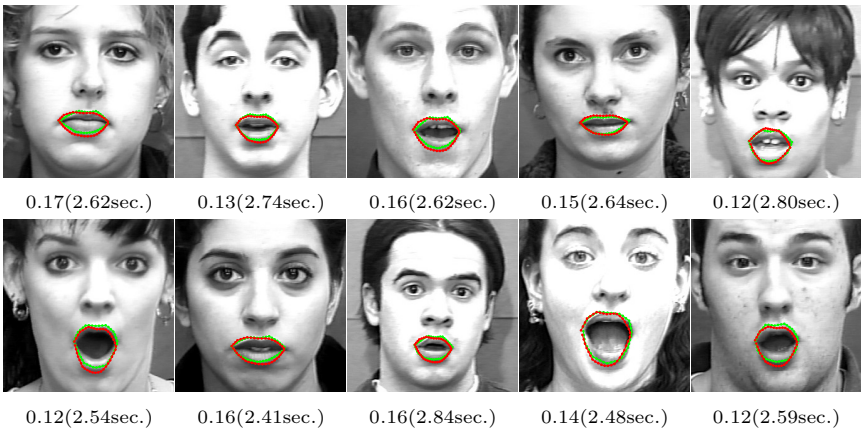


Figure 6. Test lip sequences displaying the "surprise" expression. Below each image, it is shown the Jaccard error and the running time to perform the segmentation.

Table 2. Comparison in the test sequences of the lip segmentation problem. The sequences are sorted from left to right according to Fig. 6

		Lip Sequences in Fig.6 (top row)					**Lip Sequences Metrics in Fig.6 (bottom row)**				
	Method	♯ 1	♯ 2	♯ 3	♯ 4	♯ 5	♯ 6	♯ 7	♯ 8	♯ 9	♯ 10
HDF	CAR1 [4]	12.76(3.98)	11.27(2.51)	7.29(4.67)	5.80(1.65)	5.35(1.27)	8.76(4.35)	8.95(2.45)	7.20(1.80)	11.54(3.66)	10.33(2.50)
	OurAppr.	**8.04(3.76)**	**5.05(1.23)**	**5.07(1.18)**	**5.14(0.67)**	**4.56(0.65)**	**3.49(0.62)**	**8.25(1.62)**	**5.53(1.08)**	**6.72(1.90)**	**4.74(0.40)**
MAD	CAR1 [4]	8.33(1.77)	6.98(2.89)	6.43(3.45)	5.05(1.00)	3.97(0.81)	7.22(2.75)	7.33(2.79)	6.01(1.71)	9.38(3.12)	9.43(1.91)
	OurAppr.	**6.26(2.00)**	**3.44(0.40)**	**3.77(0.62)**	**3.39(0.76)**	**3.38(0.45)**	**2.61(1.01)**	**6.07(1.22)**	**3.88(0.51)**	**5.14(1.2)**	**3.42(0.41)**
JCD	CAR1 [4]	0.21(0.03)	0.21(0.05)	0.20(0.08)	0.18(0.07)	0.13(0.02)	0.21(0.06)	0.19(0.09)	0.22(0.02)	0.22(0.03)	0.25(0.03)
	OurAppr.	**0.17(0.05)**	**0.13(0.02)**	**0.16(0.02)**	**0.15(0.02)**	**0.12(0.01)**	**0.12(0.04)**	**0.16(0.01)**	**0.16(0.01)**	**0.14(0.02)**	**0.12(0.03)**
AV	CAR1 [4]	4.20(0.81)	3.67(1.48)	3.32(1.87)	**2.49(0.47)**	**2.01(0.39)**	3.74(1.51)	**3.65(1.37)**	3.00(0.82)	4.69(1.47)	4.86(1.09)
	OurAppr.	**3.76(1.36)**	**2.14(0.53)**	**2.46(0.29)**	2.63(0.47)	2.01(0.50)	**1.91(0.31)**	4.04(0.87)	**2.44(0.33)**	**3.29(0.72)**	**2.30(0.14)**

[9] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta. Databased-guided segmentation of anatomical structures with complex appearance. In *CVPR*, 2005. 1, 2, 6, 7

[10] A. Hammoude. *Computer-assited Endocardial Border Identification from a Sequence of Two-dimensional Echocardiographic Images*. PhD thesis, University Washington, 1988. 6

[11] U. Helmke, K. Hper, P. Y. Lee, and J. Moore. Essential matrix estimation using Gauss-Newton iterations on a manifold. *Int. Journal of Comp. Vision*, 74(2):117–136, 2007. 2

[12] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 4, 6

[13] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9):850–863, 1993. 6

[14] K. Hper and J. Trumpf. Newton-like methods for numerical optimization on manifolds. In *In Proc. of the 38th Asilomar Conference on Signals, Systems and Computers*, pages 136–139, 2004. 2

[15] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008. 5

[16] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010. 6

[17] J. C. Nascimento and J. S. Marques. Robust shape tracking with multiple models in ultrasound images. *IEEE Trans. Imag. Proc.*, 17(3):392–406, 2008. 5, 6

[18] J. C. Nascimento and J. G. Silva. Manifold learning for object tracking with multiple motion dynamics. In *ECCV*, 2010. 1, 3

[19] L. Neil. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 7:455–491, 2005. 3

[20] J. A. Noble and D. Boukerroui. Ultrasound image segmentation: A survey. *IEEE Trans. Med. Imag.*, 25(8):987–1010, 2006. 2

[21] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society*, 50(5):537–544, 2003. 3

[22] J. G. Silva, J. S. Marques, and J. M. Lemos. Selecting landmark points for sparse manofold learning. In *NIPS*, 2005. 3

[23] S. Smith. Optimization techniques on Riemmanian manifolds. In A. Bloch, editor, *In Hamiltonian and gradient flows, algorithms and control*, pages 113–136. American Mathematical Society, 2004. 2

[24] Y. Tian, T. Kanade, and J. Cohn. Robust lip tracking by combining shape, color and motion. In *Proceedings of the 4th Asian Conference on Computer Vision*, pages 1040–1045, 2000. 2

[25] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Conf. Computer Vision and Pattern Rec. (CVPR)*, pages 511–518, 2001. 1

[26] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. *NIPS*, 18, 2005. 3

[27] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *IJCV*, 70(1):77–90, 2006. 3

[28] Y. Zhan, X. S. Zhou, Z. Peng, and A. Krishnan. Active scheduling of organ detection and segmentation in whole-body medical images. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pages 313–321. Springer, 2008. 1

[29] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. N. Metaxas, and X. S. Zhou. Towards robust and effective shape modeling: Sparse shape composition. *Medical image analysis*, 16(1):265–277, 2012. 1

[30] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. *IEEE Trans. Med. Imaging*, 27(11):1668–1681, 2008. 1, 2, 5, 7

[31] X. S. Zhou, D. Comaniciu, and A. Gupta. An information fusion framework for robust shape tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(1):115–129, 2005. 1, 2, 6, 7